NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MODEL FOR ESTIMATION OF THERMAL HISTORY PRODUCED BY A SINGLE PASS UNDERWATER WET WELD

by

Jay F. Dill

December, 1997

Thesis Advisor: Alan G. Fox

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. **REPORT DATE** December 1997	3. **REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
MODEL FOR ESTIMATION OF THERMAL HISTORY PRODUCED BY A SINGLE PASS UNDERWATER WET WELD

5. FUNDING NUMBERS

6. AUTHOR(S) Jay F. Dill

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)* Thermal history calculations for single pass underwater wet weldments were made by solving the appropriate heat transfer equations using the three-dimensional Crank-Nicholson finite difference method. The Adams approach, which defines the fusion line temperature as a boundary condition, was adopted. Tsai and Masubuchi's semi-empirical correlation, defining the surface heat transfer coefficient of underwater weldments, was used to determine the heat loss through the surface of the welded plate. As expected, the calculated cooling rates in heat affected zones (HAZs) of underwater wet welded ferritic steels were found to be somewhat faster than equivalent cooling rates calculated for the same weldments generated in air. However, the effect of water temperature on cooling times in the HAZ between 800° and 500°C (the parameter conventionally used to measure the cooling rate in the HAZ) was found to be minimal. These calculations suggest that HAZ microstructure of underwater wet welded ferritic steels should be independent of water temperature. This prediction was confirmed by microstructural studies of samples of ASTM A516 grade 70 steel which were underwater wet welded at water temperatures of 31°, 10° and 3° C respectively and for which similar HAZ microsructures were obtained in each case.

14. SUBJECT TERMS Underwater wet welding			15. NUMBER OF PAGES 64
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

MODEL FOR ESTIMATION OF THERMAL HISTORY PRODUCED BY A SINGLE PASS UNDERWATER WET WELD

Jay F. Dill
Lieutenant, United States Navy
B.S., United States Naval Academy, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December, 1997

ABSTRACT

Thermal history calculations for single pass underwater wet weldments were made by solving the appropriate heat transfer equations using the three-dimensional Crank-Nicholson finite difference method. The Adams approach, which defines the fusion line temperature as a boundary condition, was adopted. Tsai and Masubuchi's semi-empirical correlation, defining the surface heat transfer coefficient of underwater weldments, was used to determine the heat loss through the surface of the welded plate. As expected, the calculated cooling rates in heat affected zones (HAZs) of underwater wet welded ferritic steels were found to be somewhat faster than equivalent cooling rates calculated for the same weldments generated in air. However, the effect of water temperature on cooling times in the HAZ between 800° and 500°C (the parameter conventionally used to measure the cooling rate in the HAZ) was found to be minimal. These calculations suggest that HAZ microstructure of underwater wet welded ferritic steels should be independent of water temperature. This prediction was confirmed by microstructural studies of samples of ASTM A516 grade 70 steel which were underwater wet welded at water temperatures of 31°, 10° and 3°C respectively and for which similar HAZ microsructures were obtained in each case.

TABLE OF CONTENTS

ACKNOWLEDGMENT

I would like to express my sincere gratitude and appreciation to Professor Alan Fox for his advice, support and motivation throughout the thesis process.

I also would like to thank my wife for all the love and patience shown over the years.

I. INTRODUCTION

Reliable underwater wet welds on ferritic steels are required to reduce ship repair costs in the future. Currently the major drawback to the underwater wet welding of these steels is the problem of producing high quality welds which have no under-bead cracking in the heat affected zone (HAZ) [Ref. 1,2]. The rapid cooling rates associated with underwater weld passes results in a HAZ microstructure that is predominantly martensite, even in those steels which have a fairly low carbon equivalent and carbon content. This, combined with the high residual stresses and the high concentrations of diffusible hydrogen in the weld pool arising from dissociation of the surrounding water, means that all the requirements for hydrogen assisted under-bead cracking are easily met in underwater welding.

In the past, analytical expressions have been used to describe the thermal history of weldments made in air [Ref.1]. This approach has been improved by the development of finite difference models which rely on fewer simplifying assumptions [Ref.1] and, for single pass gas tungsten arc weldments made in air, a very accurate thermal history can be derived [Ref.1]. Unfortunately, heat transfer from the surface of a hot welded plate to surrounding water is very complex, although Tsai et al., have suggested the use of a semi-empirical correlation for the heat transfer coefficient based on the observation of bubble dynamics in the vicinity of the arc [Ref.4]. This approach will also be adopted in the present work.

In the past, in order to avoid having to model the energy density distribution of the arc, the thermal properties of the filler rod which determine the size and shape of the fusion zone, some workers (e.g., Adams and Christensen) have incorporated fusion zone

size and shape measurements as boundary conditions for the development of their heat transfer models [Ref. 1,5,10]. This is very effective since it means that many of the uncertainties associated with the solution of the welding heat transfer problem are eliminated.

In the present work, a finite difference model which predicts the time-temperature history of underwater wet weldments made on ferritic steel will be developed. This will use the fusion zone boundary condition for the solution of the resulting non-linear partial differential equation. Heat transfer to the surrounding water will be accounted for using the model of Tsai et al. described earlier in this section.

II. BACKGROUND

A. Underwater Shielded Metal Arc Welding

Underwater shielded metal arc welding is somewhat similar to SMAW performed in air. Both consist of maintaining an electric arc between the tip of a consumable electrode and the base plate, see Figure 2.1. The consumable electrode is generally coated with various materials in order to perform one or more of the following functions [Ref.1]:

1. Provide a gaseous shield to protect the molten metal from the surrounding environment.
2. Provide deoxidizers and fluxing agents to deoxidize and cleanse the weld metal.
3. Provide arc stabilizers to maintain a stable arc during welding.
4. Provide alloying elements and/or metal powders to the weld in order to control composition and increase deposition rate respectively.

The underwater wet SMAW process has the advantage, over other underwater welding processes, of using relatively simple, portable and inexpensive equipment [Ref.3].

Underwater SMAW differs from SMAW in air in several ways. First, the welder is restricted to wearing equipment to provide breathing air, which generally consists of a vision-impairing helmet. Other differences include the sole use of a direct current straight polarity power supply (DCSP) and the welding equipment being located remotely from the welding site [Ref.3].

The most notable difference between the two types of SMAW is the environment in which the heated and or molten metal is in contact during the cooling process. In both air and underwater SMAW the molten pool is surrounded, in part, by a gaseous shield resulting from the consumption of the electrode coating by the arc. When surrounded by water, this shield also consists of dissociated hydrogen and oxygen produced by the arc. The overall result is faster cooling rates in underwater wet welding since water has a

higher specific heat capacity than air and more hydrogen available for diffusion into the molten weld pool.

B. Effects of Rapid Cooling Rate on Microstructure and Mechanical Properties.

1. The Martensite Transformation

One of the problems associated with the rapid cooling rates that occur in underwater wet welding is the inevitable appearance of martensite in the coarse-grained-heat-affected-zone (CGHAZ) even when the carbon equivalent and carbon content of the steel is fairly low.

Martensite is hard and brittle because it has a high dislocation density and a body-centered tetragonal crystal structure with few slip systems. These develop during the diffusionless shear of austenite to martensite during rapid cooling. As discussed in the next section, hard, brittle microstructures are very susceptible to hydrogen assisted cracking.

The microstructure resulting from a given cooling rate can be determined with the aid of a continuous-cooling-transformation diagram [Ref.8]. Such a diagram for ASTM A516 grade 70 steel is shown in Figure 2.2. The diagram is used by identifying the cooling rate curve of interest and following it to the final temperature. The microstructure formed consists of those depicted on the diagram, as individual labeled areas, through which the cooling rate line passes. Values of the Vickers hardness are also available at the bottom of each cooling rate line. Cooling rates at the fusion line and in the CGHAZ associated with the underwater wet welding of A516 grade 70 steel inevitably lead to a predominantly martensitic structure.

2. Hydrogen Cracking and the Effect of Water Temperature in Underwater Welding.

Hydrogen cracking becomes prevalent in high carbon steels when the following four factors are present simultaneously [Ref. 1]:

1. Hydrogen present in the weld metal.
2. High stress.
3. Susceptible microstructure (Martensite).
4. Relatively low final temperature.

It has already been pointed out that the levels of hydrogen surrounding the molten weld pool are higher in underwater welding than air due to the dissociation of water into hydrogen and oxygen. High stresses are introduced as a result of the rapid heating and cooling as well as the volume change associated with the martensitic transformation and any restraint that the plates to be welded may be under. Consequently hydrogen assisted underbead cracking is almost inevitable during the underwater wet welding of ASTM A516 grade 70 steel which is the alloy of particular interest in the present work.

The process responsible for hydrogen cracking is shown in Figure 2.3. As shown, hydrogen diffuses into the molten pool and is retained in the solidified austenite. The austenite produced in the weld pool is generally lower in carbon content than that of the base metal or HAZ. The difference in carbon content causes the weld pool austenite to transform before the HAZ austenite transforms to martensite. Decomposition products such as ferrite, bainite and martensite have lower solubility and higher diffusivities for hydrogen than austenite. This results in large amounts of hydrogen diffusing across the transformation boundary, as shown, and becoming trapped to produce hydrogen assisted underbead cracking in the HAZ [Ref. 1,10].

The effect of water temperature on the extent of underbead cracking in A516 grade 70 steel is the important issue for this work and it may also be an issue for other steels with similar carbon equivalents and carbon contents. A516 grade 70 steel has been studied by West et al. [Ref.6] and Johnson [Ref.3]. In this work three weldments, each a part of an underwater qualification process, were produced in water of different temperatures and underbead cracking was found to be more prevalent in the colder water specimens. All samples were ASTM A516 grade 70 steel on which full penetration V-groove weldments were made in the horizontal position with backing bars and full restraint [Ref.6]. The first sample (UWW03) was welded in 3°C seawater at a depth of 22 feet. Underbead cracking was detected visually by the diver-welder immediately on weldment completion [Ref.6]. The sample was sectioned perpendicular to the welding direction with each section showing significant cracking running predominantly parallel to the fusion line. Crack lengths varied from 1 to 14mm and appeared to initiate in the HAZ with occasional propagation across the fusion line [Ref.6]. The second sample (UWW10) was performed in 10°C seawater at a depth of 18 feet. No cracking was detected with the naked eye in the finished weld, however sectioning showed many 1 to 8mm cracks with characteristics similar to those found in UWW03. The third sample (UWW31) was welded in 31°C freshwater at a depth of 24 feet. Cracks in this sample became apparent in the sectioned samples during optical microscopic observation at 64X [Ref.3].

West et al. [Ref.6], surmised that water type (the cracking occurred in both seawater samples and much less in the freshwater sample) was irrelevant based on previous underwater welding experience although it is well known that brine solutions

are more effective quenchants than pure water. Furthermore, subsequent testing in freshwater resulted in cracks similar to those observed in seawater. Pure hydrogen-induced cracking was also discounted by West in that a decrease in water temperature produced an increase in crack length with no apparent change in surrounding hydrogen levels. Hydrogen induced cracking usually manifests itself as small cracks parallel to the fusion zone and the observed cracking showed the quench-type crack tendency for propagation across the fusion zone into the weldment itself.

Johnson found, using a scanning electron microscope (SEM) and energy dispersive analysis of x-rays (EDX), that the weld metals of warmer temperature samples contained a larger volume fraction of slag and oxide inclusions. He therefore suggested that the inclusions acted as hydrogen sinks, thus reducing the hydrogen available to cause cracking in the warmer water samples.[Ref. 3]

C. Previous Models

Rosenthal's classic analytical solutions for two and three dimensional heat flow during welding are solutions, to this particular heat transfer problem, by which even current computational welding models still measure their accuracy [Ref.1,11]. Rosenthal's solution for three-dimensional heat diffusion for a single pass on an infinitely thick plate is:

$$\frac{2\pi k_s R(T-To)}{Q} = \frac{-U(R-x)}{2\alpha_s}$$

where

T = temperature,
To = workpiece temperature before welding,
k_S = thermal conductivity of the solid,

Q = heat input to the workpiece,
U = welding speed in the negative x direction,
α_s = thermal diffusivity of the solid,
R = radial distance from the origin $(x^2+y^2+z^2)^{1/2}$
x = the distance behind the heat source.[Ref. 1]

In order for his model to be analytical Rosenthal made the following assumptions:

1. Point heat source.
2. No melting and negligible heat of fusion.
3. Thermal properties unchanged with temperature.
4. No heat loss from the work piece surface.
5. Infinitely wide and thick work piece.

These assumptions mean that the calculations and equivalent experimental results differ somewhat. The assumption of a point heat source, for example, causes the solution to tend toward infinity as the distance from the source goes to zero, even though the power of the heat source is finite. This means that the calculations are only really reasonable in the HAZ. In addition, the assumptions of constant thermal properties, and negligible heat of fusion can, depending on the material, introduce significant errors. However, this model, although not perfect, can produce reasonable results for HAZ thermal histories that can be used to understand real welding problems.

C.M. Adams developed Rosenthal's solution so that it was particularly applicable to calculating thermal histories in the HAZ. Using the fusion line, as a boundary condition, Adams was able to calculate peak temperatures in the HAZ at a given distance from the fusion boundary on the surface of the weldment. The resulting equation for the three dimensional case is:

$$\frac{1}{Tp-To}=\frac{5.44\pi k_s\alpha_s}{QU}\left[2+\left(\frac{UY}{2\alpha_s}\right)^2\right]+\frac{1}{Tm-To}$$

where all the symbols are as before except

Tp = peak temperature reached at a distance Y from the fusion boundary,
Tm = melting temperature.

Although this method avoids the problems associated with modeling the weld pool, it still suffers from the same approximations associated with the Rosenthal solution.

Christensen et al. also exploited weld bead geometry, in their Rosenthal based model. In this approach, a solid curve, depicting relatively good agreement between calculated and experimental results, is shown in Figure 2.4. The equations, which describe a dimensionless weld depth, D, and a dimensionless operating parameter, n, are as follows:

$$D=\frac{dU}{2\alpha_s}$$

$$n=\frac{QU}{4\pi\alpha_s^2\rho C(Tm-To)}$$

where all variables are the same as above except d, ρ and C which stand for weld bead depth, material density and specific heat respectively. This approach could also be used to find the location associated with a maximum temperature T_H in the HAZ by substituting Tm with T_H. In ferritic steels T_H would typically be the austenite decomposition temperature (M_S in the case of martensite).

More recently, numerical computer models have been developed allowing the assumptions of Rosenthal to be relaxed. Numerical models have their own inherent

inaccuracies, thus are generally referred to as approximations. These can produce good solutions, but computation times are often long. The computation time is reduced often by reinstating some of the Rosenthal's assumptions or making numerical approximations of observed phenomena. For example, a model developed by Ule et al. used a Gaussian power density distribution for the source and took into account surface heat losses and the variation of the thermal properties of the welded metal; solutions using this model take a high-speed computer 4.6 minutes for every second of model time [Ref.11].

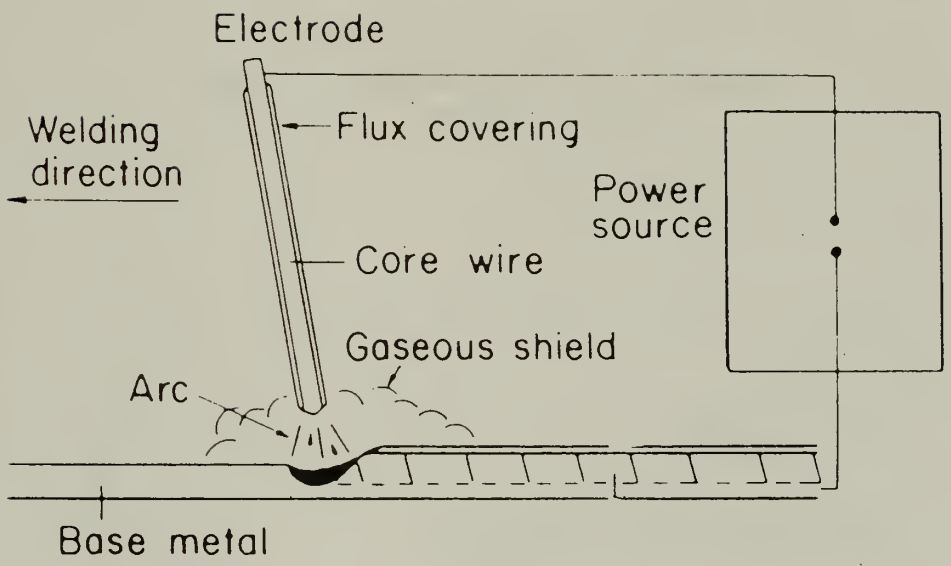

Figure 2.1 Shielded Metal Arc Welding Process [Ref.1]

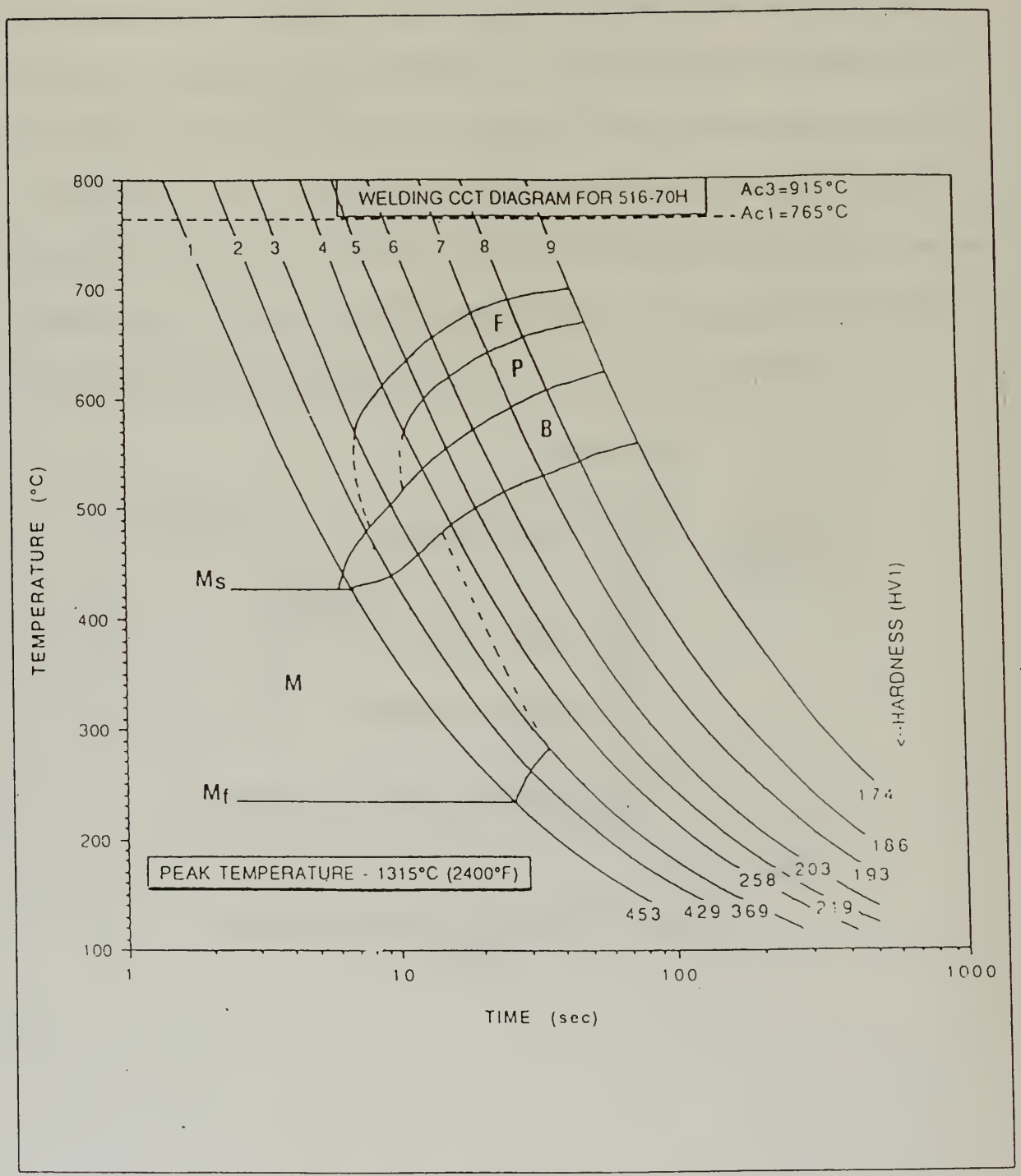

Figure 2.2 Continuous Cooling Transformation Diagram for ASTM A516 grade 70 steel [Ref.13]

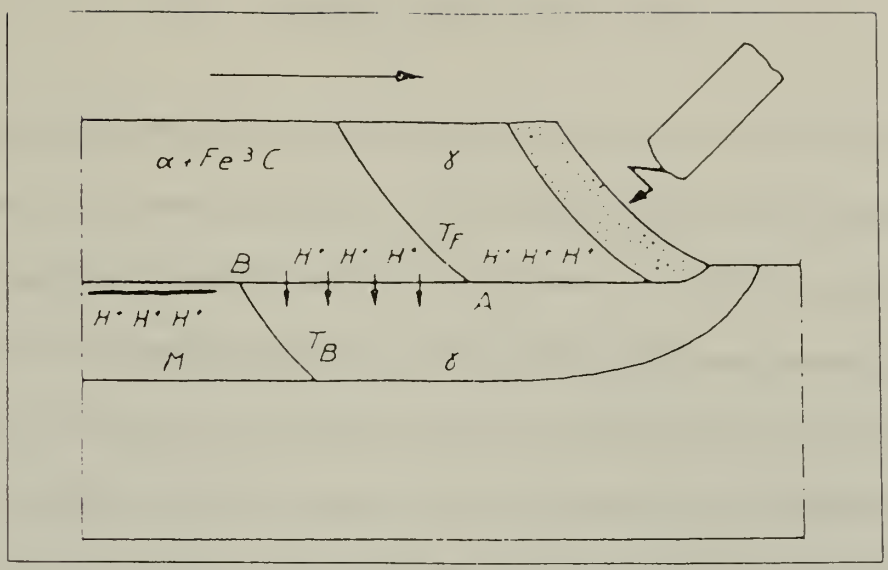

Figure 2.3 Mechanism of Hydrogen Cracking.
[Ref.1]

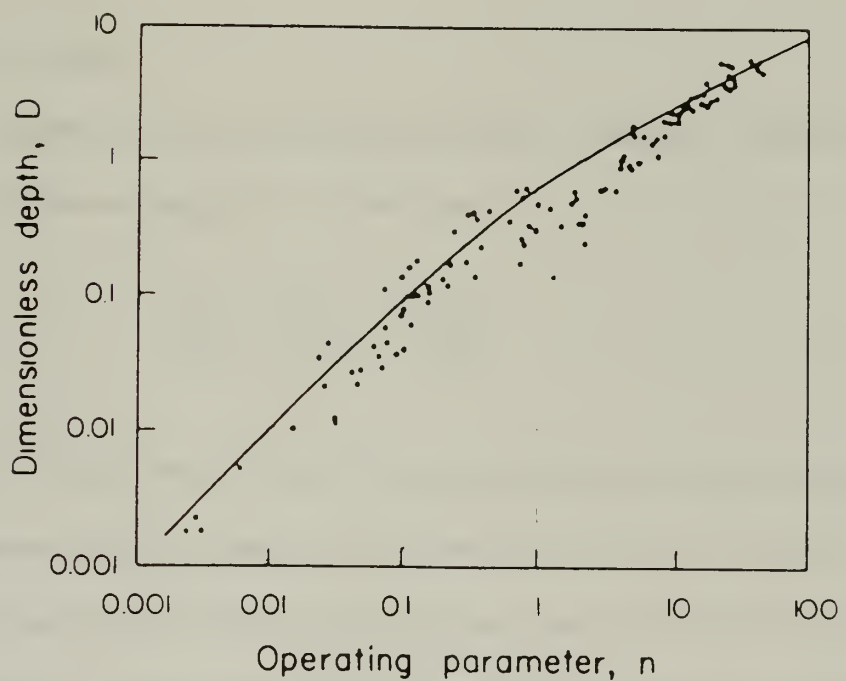

Figure 2.4 Dimensionless weld depth D as a function of dimensionless operating parameter n.
[Ref.1]

III. MODEL DEVELOPMENT

Although Rosenthal's analytical solution gives a good approximation of the HAZ thermal history during welding, little can be learned when welding in an environment in which the assumptions associated with this approach become invalid. Many times, removing a certain assumption, produces a nonlinear partial differential equation requiring the use of a numerical method. Several numerical models have been developed [Ref.1,5,11] in which a compromise between accuracy, a model's point of interest and available computational power must be decided upon. In the case of an underwater weld the heat lost through the surface of a welded plate becomes significant in that the heat transfer coefficient increases by a factor of 100 over that experienced in air [Ref. 5].

A. Problem Strategy

A three-dimensional model, utilizing the stationary coordinate, transient heat diffusion partial differential equation was developed.

$$\frac{1}{\alpha}\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}$$

This equation was chosen over that of the moving coordinate system equation due to the instability the second equation experiences in the area ahead of the weld pool [Ref. 7] and the comparative ease with which the temperature history can be extracted from the first. As such, the weld pool in this model is moved incrementally through the coordinate system at the defined weld speed, using the results of each previous step as the initial condition to the next. With the exception of no surface heat loss and point heat source,

the assumptions associated with Rosenthal's thick plate solution were used in order to provide a base for solution comparison.

1. Finite Difference Method.

The finite difference method (FDM) of numerical solution was chosen over the finite element method (FEM) for three reasons [Ref. 7]:

1. FDM is simple to formulate and requires less computational work to arrive at a solution.
2. Unlike FEM, the accuracy of FDM can be examined by order of truncation error in the Taylor series expansion.
3. FDM is easy to apply for solution to engineering problems involving simple geometry.

The Crank-Nicholson finite difference scheme was chosen for the solution to this model. Crank-Nicholson is an implicit differencing scheme developed by taking the arithmetic average of the implicit and explicit schemes, resulting in the advantages of both. These advantages include second order accuracy in both time and space with no restriction on time step size [Ref. 7].

The four node types making up the model are interior nodes, side nodes, edge nodes and corner nodes. The Crank-Nicholson finite difference discretization of the three-dimensional diffusion equation for interior nodes is listed below and applies to any node located in the interior of the model.

$$\frac{T_{i,j,k}^{n+1} - T_{i,j,k}^{n}}{\alpha \Delta t} = 1/2\left(\Delta_{xx} T_{i,j,k}^{n+1} + \Delta_{yy} T_{i,j,k}^{n+1} + \Delta_{zz} T_{i,j,k}^{n+1}\right) + 1/2\left(\Delta_{xx} T_{i,j,k}^{n} + \Delta_{yy} T_{i,j,k}^{n} + \Delta_{zz} T_{i,j,k}^{n}\right)$$

$$\Delta_{xx} T_{i,j,k}^{n} = \frac{1}{(\Delta x)^2}\left[T_{i+1,j,k}^{n} - 2T_{i,j,k}^{n} + T_{i-1,j,k}^{n}\right]$$

$$\Delta_{yy}T^n_{i,j,k} = \frac{1}{(\Delta y)^2}\left[T^n_{i,j+1,k} - 2T^n_{i,j,k} + T^n_{i,j-1,k}\right]$$

$$\Delta_{zz}T^n_{i,j,k} = \frac{1}{(\Delta z)^2}\left[T^n_{i,j,k+1} - 2T^n_{i,j,k} + T^n_{i,j,k-1}\right]$$

Side nodes are those nodes lying on any of the 6 planes defining an exterior boundary of the mesh. The discretization for the side nodes is found by performing an energy balance across the external boundary, and substituting the results into the equation for interior nodes; thus eliminating the terms relating to the nonexistent node. Edge nodes and corner nodes are those nodes that lie on the intersection of two and three boundary planes respectively. The discretization equation for these two nodes is found in the same manner as that for side nodes.

With an equation provided for each of the N nodes in the model, the solution becomes one of solving N simultaneous equations. The resulting temperature distribution, a function of the material properties, boundary conditions, initial conditions and time step, is determined. However, it must be noted that the boundary conditions change with changing temperature. The temperature distribution for each time step is found using the lagging properties method. This method consists of calculating the heat transfer coefficient at the nodes current temperature, solving for T^{n+1}, and recalculating the heat transfer coefficient. The two heat transfer coefficients are combined and T^{n+1} is solved a second time using the corrected heat transfer coefficient.[Ref. 7]

2. Base Plate

The base plate is defined using the thermal properties for ASTM A516-70 steel as listed in Table 3.1 and contains the coordinate system on which the weld pool will propagate. The axis of symmetry inherent in the welding problem was used in order to reduce the calculations required to provide a solution. The axis of symmetry is defined down the center of the weld bead and is exploited by slicing the plate lengthwise through the weld bead, as depicted in Figure 3.1, and applying a zero flux boundary condition to the resulting surface. The weld pool is then applied at the corner defined by the plate's leading edge and zero flux surface.

3. Weld Pool

As mentioned previously, the weld pool shape is indicative of the power input, arc efficiency and melting properties of the plate [Ref. 1,5]. Therefore, by measuring the weld pool shape and size and applying them to the coordinate system as a boundary condition, the properties listed above are inherently included in the solution. The weld pool shape is determined by measuring the fusion line visible in the sample and rotating it about an axis passing through its centroid thus producing a volume. This volume is then defined as accurately as possible in the coordinate system, with the associated temperature defined as the melting temperature. The weld pool volume is then propagated into and through the coordinate system as a function of the weld speed and distance between nodes in the weld direction. For the samples of interest, the fusion

zones were close enough in size that they were both modeled using the same coordinate system.

4. **Boundary Conditions**

Tsai and Masubuchi studied the phenomenon of rapid cooling in underwater welding and developed a semi-empirical equation describing the heat transfer coefficient on the plate's surface [Ref. 5].

$$\overline{h} = 675(Ts - Tw)^{1/4}$$

where

Ts = temperature of the plates surface
Tw = temperature of the surrounding water.

These results were based on a high speed motion picture study conducted at MIT, and described the phenomenon of gas bubble growth and departure from the area surrounding the welding arc as depicted in Figure 3.2. It was observed that once the bubble reached a maximum radius it departed and a new bubble began to form. The periodicity for bubble departure was 0.071 of a second, thus setting up a gas column rising away from the welding arc. The shear force, acting between the gas and liquid, acts to draw the heated water in contact with the surface surrounding the welding arc up the column only to be replaced by cooler water. The stable arc zone encompassing the heat input circle allows the assumption that no heat is lost from the plate surface inside the input circle. Tsai's correlation is thus applied, as a function of plate surface temperature, to the mesh surface outside the stable gas zone.

The heat transfer coefficients for the vertical sides and bottom of the mesh were calculated using coorelations that describe convective heat flow from a vertical surface and the bottom of a heated plate respectively. Boiling on these surfaces was neglected in

that the saturation temperature at weld depth (400°C) was considerably higher than that experienced at the weld plates boundaries. It is understood that these coorelations were developed for steady state conditions, however it is generally accepted that these coorelations be used to estimate heat flow from the transient-welding problem [Ref. 4].

During their high-speed photography examination of the underwater welding process, Tsai et al noted the absence of boiling along the surface of the plate. Boiling, in fact, was observed only in the narrow area just outside the stable gas zone while not occupied by an expanding bubble. This explains the under-estimation of heat transfer based on models which used a "boiling zone" approach.[Ref. 5]

Chemical Composition [wt%]		Thermal Properties		
C_{max}	0.31	Thermal Conductivity	60.5	[w/mK]
Mn_{max}	0.85-1.20	Thermal Diffusivity	17.7e-6	[m^2/s]
P_{max}	0.035	Melting Temperature	1800	[K]
S_{max}	0.04			
Si_{max}	0.15-0.30			

Table 3.1 Properties of ASTM A516 Grade 70 Steel

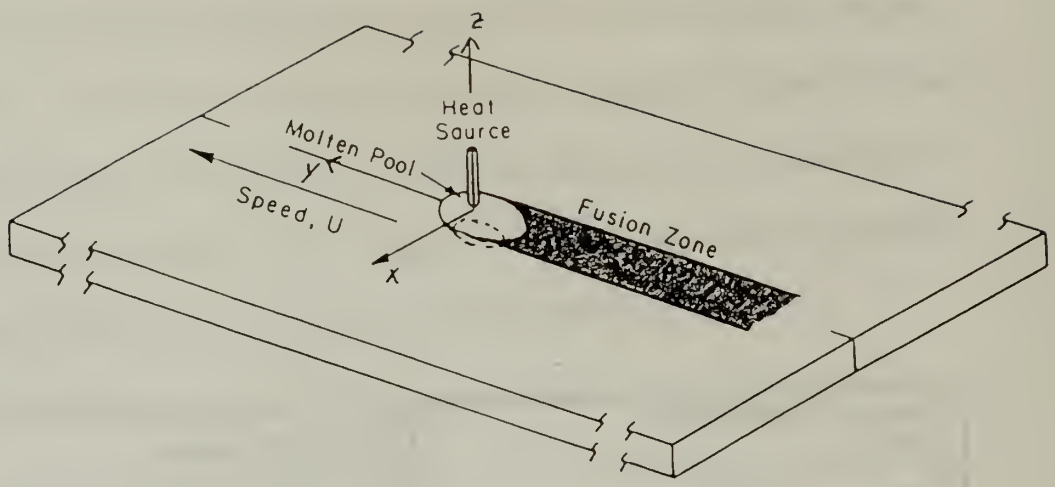

Figure 3.1 Model Diagram
[Ref.1]

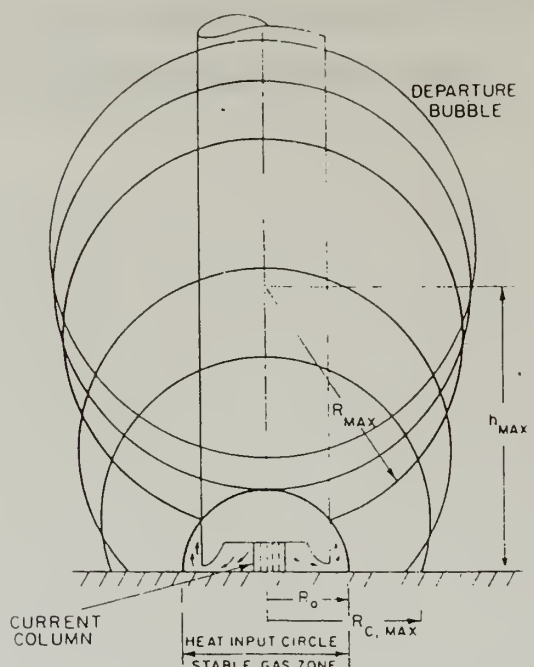

Figure 3.2 Bubble Growth and Departure from Stable Arc Area
[Ref.5]

IV. RESULTS

The computer model was run using an 1800 node mesh to describe the plate, and took approximately18 hours to run. The mesh spacing used was 3.5 mm in the x-direction (perpendicular to motion of the torch) and 1.5 mm in the y-direction. It would have been highly desirable to use a mesh in which the x-direction spacing was 1mm, but this would have required 5250 nodes to define the plate, and run time would have exceeded a reasonable limit. Greater resolution may be possible in the future through the use of a nested mesh scheme similar to that developed by Ule et al. to describe gas tungsten arc welding [Ref.11].

The program produces a matrix containing the temperature history of each node defined in the mesh. By examining the results, it was determined that end effects became negligible after approximately 5 second of welding. It was also found that the minimum plate half-width required to provide for the weldpool's temperature decaying back to that close to the surroundings was approximately 40mm. These measurements were determined using previous run data in order to reduce the dimensions of the plate and provide for a more closely spaced mesh.

The model produced as a "saw-toothed" characteristic in the region of the curve associated with the highest temperature gradient and disappeared altogether at a temperature of approximately 1000°K. This characteristic was the result of the incremental step motion of the weld pool through the coordinate grid. By reducing the size of the steps (distance between nodes in the direction of welding) these characteristics increased in number, on a given length of curve, and became less distinct.

The "saw-teeth" were used to produce a bounded area through which the real solution lies. It should be recognized that, in examining a node behind the weld pool, the time step immediately following the incremental movement of the weldpool, produced a higher cooling rate than would be experienced were the arc moving in a linear fashion. The high cooling rate experienced here produces a false low temperature for the given time period. During subsequent time step solutions, preceding the next weld pool step, the node temperature will approach a steady state solution were the weld pool to remain stationary. Of course, in this case, the weld pool moves prior to the steady state solution being reached. The steady state temperature is estimated by solving for the temperature using one extra time step, therefore producing a false high. By connecting the false highs and lows an area is defined in which the true solution lies. As the distance between nodes decreases, the area associated with the false highs and lows also decreases rendering a better solution.

Figure 4.1 shows the time history of two nodes in the model mesh, one at the plate surface fusion line and the other at the plate surface 3.5mm from the fusion line for two different surrounding water temperatures. The node history shows the fusion line node initially being exposed to the fusion line boundary condition, and expected temperature drop as the weld pool moves away. The 800°-500°C range is shown and the time to cool through this temperature difference is close to one second for all three water temperatures. It is important to note that the change in the time difference, $\Delta t_{800\text{-}500}$, experienced as a result of differing water temperatures is less than 0.1 second. This time difference is deemed negligible, in terms of altering the microstructure of the HAZ in a steel weldment. This was verified by the optical microscope observations carried out on

the underwater wet-welded ASTM A516 grade 70 steel samples by Johnson [Ref.3]. All three samples consisted of virtually identical CGHAZ microstructures made up mostly of martensite and a small amount bainite [Ref. 3] despite their being welded at different water temperatures. Vickers hardness readings, in the area of the fusion line, were well above 450 in each case [Ref.3], and this is exactly what is expected for the $\Delta t_{800\text{-}500}$ values predicted by the model and used in conjunction with the welding-continuous-cooling-transformation-diagram for ASTM A516 grade 70 steel (Figure 2.2). A summary of the $\Delta t_{800\text{-}500}$ values are provided in Table 4.1, and a comparison with Rosenthal's solution using parameters, similar to those of the present work, is made. Rosenthal's solution gives a $\Delta t_{800\text{-}500}$ which is about 2.4 times greater than the present underwater numerical solutions at the fusion line and is therefore not useable for predicting thermal histories of underwater wet weldments. Tsai et al. ran their model on a thinner plate, with a slower weld-speed and higher power input and monitored a node 1mm from the fusion line. All these differences will mean greater values of $\Delta t_{800\text{-}500}$, and, indeed, their calculated $\Delta t_{800\text{-}500}$ are about 50% greater than those of the present work as shown in Table 4.1 [Ref. 12].

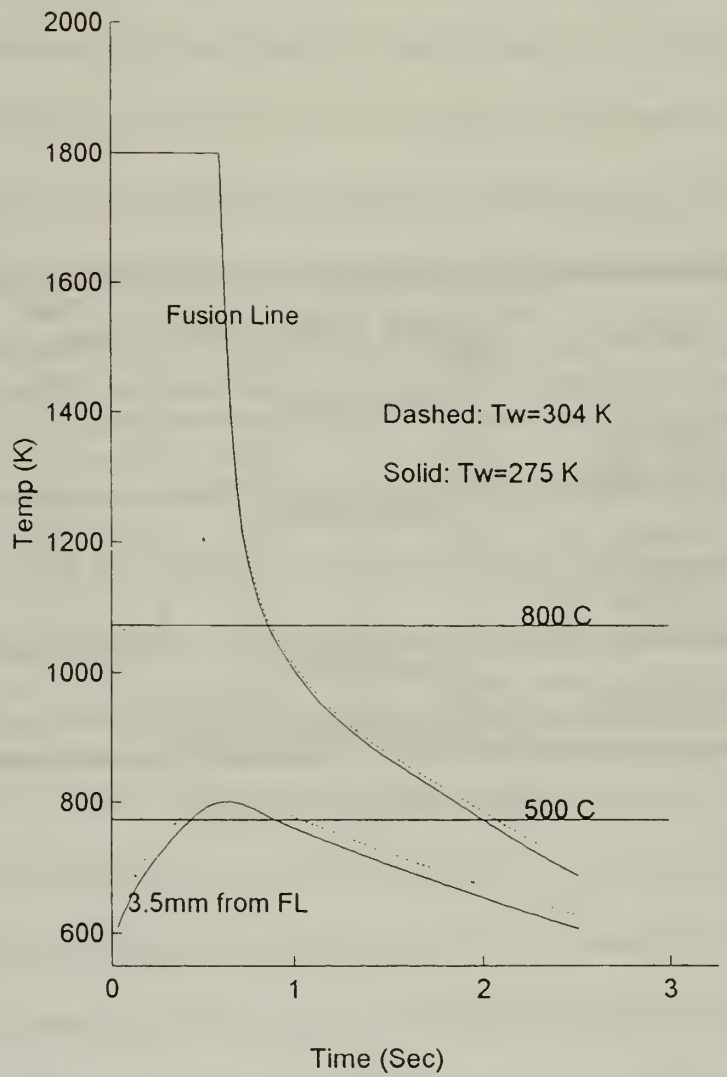

Figure 4.1 Results of the Model of the Present Work

Water Temperature [C]	Position	Weld Speed [ipm]	Power Input [kW]	Plate Thickness [in]	$\Delta t_{800\text{-}500}$
Current Model					
2	Fusion Line	6.5	3.9	¾	1.14
31	Fusion Line	6.5	3.9	¾	1.20
Tsai et al. Model					
0	1mm from Fusion Line	9	5.0	1/8	1.85
27	1mm from Fusion Line	9	5.0	1/8	1.90
Rosenthal's Model					
20*	Fusion Line	6.5	3.9	∞	2.38
* Plate Temperature prior to welding.					

Table 4.1 Model Comparisons.

V. SUMMARY

A. Conclusion

A model describing the three-dimensional heat transfer in a single pass underwater wet weld was developed using the Crank-Nicholson finite difference method. The model makes use of Tsai and Masubuchi's semi-empirical correlation describing the heat transfer coefficient on the surface of a welded plate. The program can be used to generate thermal histories, produced by welding, in a plate composed of any metal provided that the fusion zone dimensions, material properties and welding speed are known. The results produced in this case were verified using both calculated results from other available models and the observed microstructure of several underwater wet weld samples. The model results were found to be in good agreement with both standards.

Once verified, the model was run several times with different values for the surrounding water temperature. It was found that varying the surrounding water temperature by as much as 31°C produced less than a 1% change in the time required to transverse the 800°-500°C temperature range ($\Delta t_{800\text{-}500}$); thus appears to have a negligible effect on the microstructure found in the CGHAZ and is most likely not the cause of the increased cracking observed as the water temperature is lowered. This is in agreement with Johnson's findings [Ref.3] and adds credibility to the hypothesis that the observed underbead cracking is related to inclusion volume fraction vice surrounding water temperature.

B. Recommendations for Future Study.

1. More underwater weldments need to be made in order to increase the statistical significance of a cracking occurrence in samples of the same

surrounding water temperature. Also, samples should be produced in such a way so as to eliminate unwanted variables. For example, water chemistry, weld depth and the welder-diver should all be carefully controlled to be the same, if possible.

2. The possibility that the increased volume fractions of non-metallic oxide inclusions found in the samples produced in higher water temperatures act as hydrogen sinks should be investigated. If inclusions do behave in this way, it is important to understand the process.

APPENDIX

%%% MAIN1 %%%

```
%       This MATLAB script file initializes all variables, defines the welding pool, builds
%the coefficient and constant matrices associated with the finite element discretization of
%the 3-D transient heat transfer equation. It also iterates, in order to account for Tsai's
%nonlinear boundary conditions associated with the heat loss from the surface of an
%underwater SMAW on plate. The program then calculates the time period between
%weld pool moves and calls functions that move the pool in the prescribed direction.

clear

%C,R,D are the number of columns, rows and depth of the 3-D %mesh.
%Note columns are numbered left to right starting with the top 'layer' of nodes.
%Subsequent rows follow the same pattern. As an example using the C,R and D listed
%below: top layer, first row is numbered 1-10, second row 11-20. The second layer
%begins with node number 251, and the four remaining layers numbered in the same
%manner.
C=10; R=25; D=6;

%Problem parameters
Alpha=17.7e-6;
weld_speed = 0.0023;
move_time = 0;
t_count=0;delta_t=0.03;
delta_x=0.004; delta_y=0.0015; delta_z=0.003;
k=60.5; Tinf=304; To=304;
POOL_TEMP=1800;

%The vector 'parameters' is used as a global variable in order to efficiently pass values
%required of many functions.
PARAMETERS(1)=R; PARAMETERS(2)=C; PARAMETERS(3)=D;
PARAMETERS(4)=Alpha; PARAMETERS(5)=delta_t; PARAMETERS(6)=delta_x;
PARAMETERS(7)=delta_y; PARAMETERS(8)=delta_z; PARAMETERS(9)=k;
PARAMETERS(10)=POOL_TEMP; PARAMETERS(11)=Tinf;

%Posit matrix accounts for the position, within the mesh, of each non-interior node.
POSIT_MAT = Posit(C,R,D);

%Temp is a matrix in which the columns contain all the node temperatures for a given
time step and the rows contain the temperature history of the associated node.
Temp=To.*ones(R*C*D,1);
NODE_TEMPS = To.*ones(1,R*C*D);

%Pool matrix is the position of all the nodes contained in the in the weld pool
```

```
POOL_MAT = [9,10,19,20,29,30,259,260,269,270,279,280]';
POOL_MAT(:,2)= POOL_TEMP.*ones(size(POOL_MAT,2),1);

%determine the interval required between weld pool moves
t_count=t_count+delta_t;
move_delta = delta_y/weld_speed;
move_time = move_time + move_delta;

%Loop while the weld pool propogates through the mesh
while ~isempty(POOL_MAT)
        tic

        %Build coefficient and constant matrices
        %[Coef_Mat]*[Node_Temps]=[Cnst_Vec] and calculate the first estimate for
        %Tn+1
        COEF_MAT =
Bld_Cof(POSIT_MAT,POOL_MAT,PARAMETERS,NODE_TEMPS);
        CNST_VEC =
Bld_Con(PARAMETERS,NODE_TEMPS,POOL_MAT,POSIT_MAT);
        NODE_TEMPS = real(inv(COEF_MAT)*CNST_VEC);

        %Iterate until the temperature guessed for Tn+1 is less than 20 degrees from
        %calculated value for Tn+1.
        NEXT_TEMP_GUESS = NODE_TEMPS;
        flag=1;
        if flag==1
                COEF_MAT =
        Bld_Cof(POSIT_MAT,POOL_MAT,PARAMETERS,NEXT_TEMP_GUESS);
                %Use of 'real' due to occasional introduction of zero magnitude imaginary
                %parts added to all values.
                NEXT_NODE_TEMP = real(inv(COEF_MAT)*CNST_VEC);
                flag=2;
                if isempty(abs(NEXT_TEMP_GUESS  - NEXT_NODE_TEMP)>20)
                        NODE_TEMPS = NEXT_NODE_TEMP;
                else
                        NEXT_TEMP_GUESS = NEXT_NODE_TEMP;
                end
        end


        %Store the current time step solution in the correct column within Temp.
        Temp(:,size(Temp,2)+1)= NODE_TEMPS;
        t_count=t_count+delta_t

        %Check to see if the correct time step has occurred in which to move the weld
        %pool.
```

```
        %Note, the weld pool is imposed and moves down the right hand end of the
        %mesh.
        if t_count >= move_time
                POOL_MAT = Mov_Pool(PARAMETERS,POOL_MAT);
                move_time = move_time + move_delta;
        end
        toc
end
```

%%% POSIT %%%

```
function NODE_POSIT_ARRAY = Posit(C,R,D)

%This function builds a two column matrix called %NODE_POSIT_ARRAY. The first
%column contains the node numbers of all non-interior nodes. The second column
%contains the associated position number: 1-6 'Side', 11-16 %'Edge', 123-456 'Corner'.

% C=#of nodes in x-dir
% R=# of nodes in y-dir
% D=# of nodes in z-dir
% lvl_mul = # of nodes in a layer
% lvl = layer # of current node
% lvl_min = smallest node # in current layer
% lvl_max = largest node # in current layer

        LVL_MUL=R*C;
        i=1;
        boundary_flag=0;

        for N=1:D*C*R
                LVL=ceil(N./LVL_MUL);
                LVL_MIN=LVL_MUL*(LVL-1)+1;
                LVL_MAX=LVL_MIN+LVL_MUL-1;

                %Nodes located in the top layer (side #1)
                if LVL==1
                        NODE_POSIT_ARRAY(i,1)=N;
                        NODE_POSIT_ARRAY(i,2)=1;
                        boundary_flag=1;
                end

                %Nodes located on back side (side #2)
                if N > (LVL_MAX - R)
                        NODE_POSIT_ARRAY(i,1)=N;
```

```
NODE_POSIT_ARRAY(i,2)=(NODE_POSIT_ARRAY(i,2)*10+2);
                boundary_flag=1;
        end

        %Nodes located on right side (side#3)
        if rem((N),C)==0
                NODE_POSIT_ARRAY(i,1)=N;

NODE_POSIT_ARRAY(i,2)=(NODE_POSIT_ARRAY(i,2)*10+3);
                boundary_flag=1;
        end

        %Nodes located on bottom (side#4)
        if LVL==D
                NODE_POSIT_ARRAY(i,1)=N;

NODE_POSIT_ARRAY(i,2)=(NODE_POSIT_ARRAY(i,2)*10+4);
                boundary_flag=1;
        end

        %Nodes located on front (side#5)
        if N <= (LVL_MIN+C-1)
                NODE_POSIT_ARRAY(i,1)=N;

NODE_POSIT_ARRAY(i,2)=(NODE_POSIT_ARRAY(i,2)*10+5);
                boundary_flag=1;
        end

        %Nodes located on left (side#6)
        if rem((N-1),C)==0
                NODE_POSIT_ARRAY(i,1)=N;

NODE_POSIT_ARRAY(i,2)=(NODE_POSIT_ARRAY(i,2)*10+6);
                boundary_flag=1;
        end

        if boundary_flag==1
                i=i+1;
                boundary_flag=0;
        end

end
```

%%% BLD_COF %%%

```
function
COEF_MAT=Bld_Cof(POSIT_MAT,POOL_MAT,PARAMETERS,NODE_TEMPS)

%       This function accepts the appropriate parameters and builds the coefficient row
%for the associated node. The row is returned to the main program in the form of a
%vector.

COEF_ROW = [];
INDEX = [];
R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
NODES=R*C*D;

for N=1:NODES

%Check current node for boundary condition, if true apply condition.
INDEX=Find_Num(POOL_MAT,N);
if ~isempty(INDEX)
        COEF_ROW = Bnd_Cof(INDEX,POOL_MAT,PARAMETERS);

else

%Check current node for interior node and, if true, calculate appropriate row.
        INDEX=Find_Num(POSIT_MAT,N);
        if isempty(INDEX)

                COEF_ROW = Int_Cof(N,PARAMETERS,NODE_TEMPS);
        else

                POSITION=POSIT_MAT(INDEX,2);

                %Check current node for side node, if true compute appropriate row.
                if POSITION < 10
                        COEF_ROW =
Sid_Cof(N,PARAMETERS,NODE_TEMPS,POSITION);
                else
                        %Check current node for corner, if true compute appropriate row.
                        if POSITION > 100
                                COEF_ROW = Cst_Bnd(N,PARAMETERS,POSITION);
                        %Current node is edge node, if true compute appropriate row.
                        else
                        COEF_ROW =
Edg_Cof(N,PARAMETERS,NODE_TEMPS,POSITION);
```

```
                end
            end
        end
end

%Insert the computed row in the appropriate position within the coefficient matrix.
COEF_MAT(N,:)=COEF_ROW;
end
```

%%% BLD_CON %%%

```
function
CNST_VEC=Bld_Con(PARAMETERS,NODE_TEMPS,POOL_MAT,POSIT_MAT)

%This function accepts the appropriate parameters and calculates the constants for all the
%nodes in the mesh. The values are returned to the main program in the form of a vector.

TEMP_CNST = [];
INDEX = [];
R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);

Tinf = PARAMETERS(11);
NODES=R*C*D;

%Loop through all nodes in the mesh.
for N=1:NODES

%Check current node for pool boundary condition by searching for current node number
%in the matrix containing all nodes in the weld pool.

INDEX=Find_Num(POOL_MAT,N);
if ~isempty(INDEX)
        TEMP_CNST = Bnd_Con(INDEX,POOL_MAT);
else

%Check current node for interior node
        INDEX=Find_Num(POSIT_MAT,N);
        if isempty(INDEX)
                TEMP_CNST = Int_Con(N,PARAMETERS,NODE_TEMPS);
        else

                POSITION=POSIT_MAT(INDEX,2);
```

```
            %Check current node for side node
            if POSITION < 10
                TEMP_CNST =
Sid_Con(N,PARAMETERS,NODE_TEMPS,POSITION);
            else
                %Check current node for corner node
                if POSITION > 100
                    TEMP_CNST=Tinf;
                %Edge node
                else
                TEMP_CNST =
Edg_Con(N,PARAMETERS,NODE_TEMPS,POSITION);
                end
            end
        end
end
CNST_VEC(N,1)=TEMP_CNST;
end
```

%%% MOV_POOL %%%

```
function POOL_MAT = Mov_Pool(PARAMETERS,POOL_MAT)

% C=#of nodes in x-dir
% R=# of nodes in y-dir
% D=# of nodes in z-dir
% lvl_mul = # of nodes in a layer
% lvl = layer # of current node
% lvl_min = smallest node # in current layer
% lvl_max = largest node # in current layer

%This function takes the current weld pool matrix and moves it forward by one node in
%the direction of welding.

R = PARAMETERS(1);
C = PARAMETERS(2);
COUNT = 1;
TEMP_MAT = [];
LVL_MUL = R*C;

%Step through the pool matrix
for i=1:size(POOL_MAT,1)
```

```
        N = POOL_MAT(i,1);
        NT =   N + C;
        if ceil(N./LVL_MUL)== ceil(NT./LVL_MUL)
                TEMP_MAT(COUNT,1)= NT;
                TEMP_MAT(COUNT,2)= POOL_MAT(i,2);
                COUNT=COUNT + 1;
        end
end

%Return updated pool matrix.
POOL_MAT=TEMP_MAT;
```

%%% FIND_NUM %%%

```
function X = Find_Num(MATRIX,NUM)

%Searches the first column of MATRIX for NUM and returns the row number that NUM
%occupies and returns that value.

B=MATRIX(:,1);
i=size(MATRIX,1);
X=[];

%Loop through the matrix passed.
for j=1:i
        if B(j)==NUM, X=j;, end
end
```

%%% BND_COF %%%

```
function COEF_ROW = Bnd_Cof(INDEX,POOL_MAT,PARAMETERS)

%This function takes the node number associated with a node located in the weld pool
%matrix, and generates a coefficient matrix row with all zeros except a one in the
%position associated with the node of interest.

R = PARAMETERS(1);
C= PARAMETERS(2);
D= PARAMETERS(3);

NUM_NODES = R*C*D;
COEF_ROW = zeros(1,NUM_NODES);
```

```
COLUMN = POOL_MAT(INDEX,1);
COEF_ROW(COLUMN) = 1;
```

%%% INT_COF %%%

```
function COEF_ROW = Int_Cof(N, PARAMETERS, NODE_TEMPS)

%This function takes an interior node number and generates the associated row in the
%coefficient matrix.

R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
ALPHA = PARAMETERS(4);
delta_t = PARAMETERS(5);
delta_x = PARAMETERS(6);
delta_y = PARAMETERS(7);
delta_z = PARAMETERS(8);

LAYER = R*C;
T_INDEX=N;
T1_INDEX = N-LAYER; T2_INDEX = N+C; T3_INDEX = N+1;
T4_INDEX = N+LAYER; T5_INDEX = N-C; T6_INDEX = N-1;
NODES=R*C*D;
COEF_ROW = zeros(1,NODES);

COEF_ROW(T_INDEX) = 1./(ALPHA*delta_t) + 1./(delta_x).^2 + 1./(delta_y).^2 +
1./(delta_z).^2;
COEF_ROW(T1_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T2_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T3_INDEX) = -1/(2*delta_x.^2);
COEF_ROW(T4_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T5_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T6_INDEX) = -1/(2*delta_x.^2);
```

%%% SID COF %%%

```
function COEF_ROW = Sid_Cof(N, PARAMETERS, NODE_TEMPS, POSITION)

%This function receives a side node number, related position number and temperatures
%associated with all nodes in the mesh, and generates the associated coefficient row.

R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
ALPHA = PARAMETERS(4);
delta_t = PARAMETERS(5);
delta_x = PARAMETERS(6);
delta_y = PARAMETERS(7);
delta_z = PARAMETERS(8);
k = PARAMETERS(9);
Tinf = PARAMETERS(11);

LAYER = R*C;
T_INDEX=N;
T1_INDEX = N-LAYER; T2_INDEX = N+C; T3_INDEX = N+1;
T4_INDEX = N+LAYER; T5_INDEX = N-C; T6_INDEX = N-1;
NODES=R*C*D;
COEF_ROW = zeros(1,NODES);

%Check to see if current node is located on side #1.
if POSITION == 1
h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
COEF_ROW(T_INDEX) = 1./(ALPHA*delta_t) + 1./(delta_x).^2 + 1./(delta_y).^2 +
        1./(delta_z).^2 + h/(delta_z*k);
COEF_ROW(T2_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T3_INDEX) = -1/(2*delta_x.^2);
COEF_ROW(T4_INDEX) = -1/(delta_z.^2);
COEF_ROW(T5_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T6_INDEX) = -1/(2*delta_x.^2);

%Check to see if current node is located on side #2.
elseif POSITION == 2
h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
COEF_ROW(T_INDEX) = 1./(ALPHA*delta_t) + 1./(delta_x).^2 + 1./(delta_y).^2 +
        1./(delta_z).^2 + h/(delta_z*k);
COEF_ROW(T1_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T3_INDEX) = -1/(2*delta_x.^2);
COEF_ROW(T4_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T5_INDEX) = -1/(delta_y.^2);
```

```
COEF_ROW(T6_INDEX) = -1/(2*delta_x.^2);

%Check to see if current node is located on side #3.
elseif POSITION == 3
COEF_ROW(T_INDEX) = 1./(ALPHA*delta_t) + 1./(delta_x).^2 + 1./(delta_y).^2 +
        1./(delta_z).^2;
COEF_ROW(T1_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T2_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T4_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T5_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T6_INDEX) = -1/(delta_x.^2);

%Check to see if current node is located on side #4.
elseif POSITION == 4
h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
COEF_ROW(T_INDEX) = 1./(ALPHA*delta_t) + 1./(delta_x).^2 + 1./(delta_y).^2 +
        1./(delta_z).^2 + h/(delta_z*k);
COEF_ROW(T1_INDEX) = -1/(delta_z.^2);
COEF_ROW(T2_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T3_INDEX) = -1/(2*delta_x.^2);
COEF_ROW(T5_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T6_INDEX) = -1/(2*delta_x.^2);

%Check to see if current node is located on side #5.
elseif POSITION == 5h =
Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
COEF_ROW(T_INDEX) = 1./(ALPHA*delta_t) + 1./(delta_x).^2 + 1./(delta_y).^2 +
        1./(delta_z).^2 + h/(delta_z*k);
COEF_ROW(T1_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T2_INDEX) = -1/(delta_y.^2);
COEF_ROW(T3_INDEX) = -1/(2*delta_x.^2);
COEF_ROW(T4_INDEX) = -1/(2*delta_z.^2);
COEF_ROW(T6_INDEX) = -1/(2*delta_x.^2);

%Side #6 is set to a constant (surrounding water temperature) in order to shorten the
%overall computation time.
else
COEF_ROW = Cst_Bnd(N,PARAMETERS,POSITION);

end
```

%%% CST BND %%%

```
function [COEF_ROW,TEMP_CNST]=Cst_Bnd(N,PARAMETERS,POSITION)

%This function takes a node, that will maintain a constant value, and returns a coefficient
%row in which all values are set to zero except the position associated with the node of
%interest. A one is inserted in this position. The function also calculates the value for the
%constant vector.

R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
Tinf = PARAMETERS(11);

NUM_NODES=R*C*D;

COEF_ROW=zeros(1,NUM_NODES);
COEF_ROW(N) = 1;
TEMP_CNST=Tinf;
```

%%% EDG COF %%%

```
function COEF_ROW = Edg_Cof(N, PARAMETERS, NODE_TEMPS, POSITION)

%This function receives an edge node number, related position number and temperatures
%associated with all nodes in the mesh, and generates the associated coefficient row.

R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
ALPHA = PARAMETERS(4);
delta_t = PARAMETERS(5);
delta_x = PARAMETERS(6);
delta_y = PARAMETERS(7);
delta_z = PARAMETERS(8);
k = PARAMETERS(9);
Tinf = PARAMETERS(11);

LAYER = R*C;
T_INDEX=N;
T1_INDEX = N-LAYER; T2_INDEX = N+C; T3_INDEX = N+1;
T4_INDEX = N+LAYER; T5_INDEX = N-C; T6_INDEX = N-1;
NODES=R*C*D;
```

```
COEF_ROW = zeros(1,NODES);

% Edge between sides #1 and #3.
if POSITION == 13

h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
COEF_ROW(T_INDEX) = 1./(ALPHA*delta_t) + 1./(delta_x).^2 + 1./(delta_y).^2 +
1./(delta_z).^2 + h/(delta_z*k) + h/(delta_x*k);
COEF_ROW(T2_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T4_INDEX) = -1/(delta_z.^2);
COEF_ROW(T5_INDEX) = -1/(2*delta_y.^2);
COEF_ROW(T6_INDEX) = -1/(delta_x.^2);

else

COEF_ROW = Cst_Bnd(N,PARAMETERS,POSITION);

end
```

%%% BND CON %%%

```
function TEMP_CONST = Bnd_Con(N,POOL_MAT)

%This function returns a value, associated with the node in %question, for use in the
constant matrix.

. TEMP_CONST = POOL_MAT(N,2);
```

%%% EDG CON %%%

```
function TEMP_CNST = Edg_Con(N, PARAMETERS, NODE_TEMPS, POSITION)

%This function receives an edge node number, related position number and temperatures
%associated with all nodes in the mesh, and calculates the corresponding constant for the
%constant vector.

R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
ALPHA = PARAMETERS(4);
delta_t = PARAMETERS(5);
delta_x = PARAMETERS(6);
delta_y = PARAMETERS(7);
delta_z = PARAMETERS(8);
k = PARAMETERS(9);
Tinf = PARAMETERS(11);

LAYER=R*C;

T_INDEX=N;
T1_INDEX = N-LAYER; T2_INDEX = N+C; T3_INDEX = N+1;
T4_INDEX = N+LAYER; T5_INDEX = N-C; T6_INDEX = N-1;


if POSITION == 13

h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
A = (1./(ALPHA*delta_t) - 1./(delta_x).^2 - 1./(delta_y).^2 -        1./(delta_z).^2 -
h/(delta_z*k)- h/(delta_x*k))*NODE_TEMPS(T_INDEX);
B =  2*h/(delta_z*k)*Tinf + 1/(2*delta_y.^2)*NODE_TEMPS(T2_INDEX);
C =  2*h/(delta_x*k)*Tinf + 1/(2*delta_z.^2)*NODE_TEMPS(T4_INDEX);
D =  1/(2*delta_y.^2)*NODE_TEMPS(T5_INDEX) +
        1/(2*delta_x.^2)*NODE_TEMPS(T6_INDEX);
TEMP_CNST = A+B+C+D;

else
TEMP_CNST = Tinf;

end
```

%%% GET H %%%

```
function h = Get_h(T,PARAMETERS,POSITION)

%This function calculates the heat transfer coefficient associated with the position of the
%node of interest.

Tinf = PARAMETERS(11);
T_pool = PARAMETERS(10);
D = PARAMETERS(3);
delta_x = PARAMETERS(6);
delta_y = PARAMETERS(7);
delta_z = PARAMETERS(8);
rho = 1/1002;
Pr = -0.2246*T + 79.798;
beta = 12.203e-6*T - 3.4011e-3;
mu = -30.8e-6*T + 10.163e-3;
Cp = -1.2245*T+4551.5;
k = 1.601e-3*T + 0.1316;
alpha = k/(rho*Cp);
nu = (mu*9.86)/rho;
h_high=0;
h_low=0;

%check to see if node is located on side #1.
if POSITION == 1

h = 4440*(T-Tinf)^0.25;

%Check to see if node is located on side #4
elseif POSITION == 4
        Leff = (delta_x*delta_y)/(2*delta_x+2*delta_y);
        Ra = (9.86*beta*(T-Tinf)*Leff^3)/(nu*alpha);
        NuL = 0.27*Ra^0.25;
h = (NuL*k)/Leff;

%If neither of the above, node is a side node.
else
        Leff = D*delta_z;
        Ra = (9.86*beta*(T-Tinf)*Leff^3)/(nu*alpha);
        NuL = 0.68 + (0.67*Ra^0.25)/(1+(0.492/Pr)^(9/16))^(4/9);
h = (NuL*k)/Leff;
```

```
end
```

%%% INT CON %%%

```
function TEMP_CNST = Int_Con(N, PARAMETERS, NODE_TEMPS)

%This function receives an interior node number, temperatures associated with all nodes
%in the mesh, and calculates the corresponding constant for the constant vector.

R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
ALPHA = PARAMETERS(4);
delta_t = PARAMETERS(5);
delta_x = PARAMETERS(6);
delta_y = PARAMETERS(7);
delta_z = PARAMETERS(8);

LAYER=R*C;

T_INDEX=N;
T1_INDEX = N-LAYER; T2_INDEX = N+C; T3_INDEX = N+1;
T4_INDEX = N+LAYER; T5_INDEX = N-C; T6_INDEX = N-1;

A = (1./(ALPHA*delta_t) - 1./(delta_x).^2 - 1./(delta_y).^2 -
1./(delta_z).^2)*NODE_TEMPS(T_INDEX);

B =  1/(2*delta_z.^2)*NODE_TEMPS(T1_INDEX) +
1/(2*delta_y.^2)*NODE_TEMPS(T2_INDEX);

C =  1/(2*delta_x.^2)*NODE_TEMPS(T3_INDEX) +
1/(2*delta_z.^2)*NODE_TEMPS(T4_INDEX);

D =  1/(2*delta_y.^2)*NODE_TEMPS(T5_INDEX) +
1/(2*delta_x.^2)*NODE_TEMPS(T6_INDEX);

TEMP_CNST = A+B+C+D;
```

%%% SID CON %%%

```
function TEMP_CNST = Sid_Con(N, PARAMETERS, NODE_TEMPS, POSITION)
```

```
%This function receives an side node number, related position number and temperatures
%associated with all nodes in the mesh, and calculates the corresponding constant for the
%constant vector.

R = PARAMETERS(1);
C = PARAMETERS(2);
D = PARAMETERS(3);
ALPHA = PARAMETERS(4);
delta_t = PARAMETERS(5);
delta_x = PARAMETERS(6);
delta_y = PARAMETERS(7);
delta_z = PARAMETERS(8);
k = PARAMETERS(9);
Tinf = PARAMETERS(11);

LAYER=R*C;


T_INDEX=N;
T1_INDEX = N-LAYER; T2_INDEX = N+C; T3_INDEX = N+1;
T4_INDEX = N+LAYER; T5_INDEX = N-C; T6_INDEX = N-1;

%check to see if node is located on side #1.
if POSITION == 1
h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
A = (1./(ALPHA*delta_t) - 1./(delta_x).^2 - 1./(delta_y).^2 -        1./(delta_z).^2 -
h/(delta_z*k))*NODE_TEMPS(T_INDEX);
B =  2*h/(delta_z*k)*Tinf + 1/(2*delta_y.^2)*NODE_TEMPS(T2_INDEX);
C =  1/(2*delta_x.^2)*NODE_TEMPS(T3_INDEX) +
        1/(delta_z.^2)*NODE_TEMPS(T4_INDEX);
D =  1/(2*delta_y.^2)*NODE_TEMPS(T5_INDEX) +
        1/(2*delta_x.^2)*NODE_TEMPS(T6_INDEX);
TEMP_CNST = A+B+C+D;

%check to see if node is located on side #2.
elseif POSITION == 2
h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
A = (1./(ALPHA*delta_t) - 1./(delta_x).^2 - 1./(delta_y).^2 -        1./(delta_z).^2 -
h/(delta_y*k))*NODE_TEMPS(T_INDEX);
B = 1/(2*delta_z.^2)*NODE_TEMPS(T1_INDEX) + 2*h/(delta_y*k)*Tinf ;
C =  1/(2*delta_x.^2)*NODE_TEMPS(T3_INDEX) +
        1/(2*delta_z.^2)*NODE_TEMPS(T4_INDEX);
D =  1/(delta_y.^2)*NODE_TEMPS(T5_INDEX) +
        1/(2*delta_x.^2)*NODE_TEMPS(T6_INDEX);
TEMP_CNST = A+B+C+D;
```

```
%check to see if node is located on side #3.
elseif POSITION == 3

A = (1./(ALPHA*delta_t) - 1./(delta_x).^2 - 1./(delta_y).^2 -
        1./(delta_z).^2)*NODE_TEMPS(T_INDEX);
B = 1/(2*delta_z.^2)*NODE_TEMPS(T1_INDEX)+
        1/(2*delta_y.^2)*NODE_TEMPS(T2_INDEX);
C = 1/(2*delta_z.^2)*NODE_TEMPS(T4_INDEX);
D =  1/(2*delta_y.^2)*NODE_TEMPS(T5_INDEX) +
        1/(delta_x.^2)*NODE_TEMPS(T6_INDEX);
TEMP_CNST = A+B+C+D;

%check to see if node is located on side #4.
elseif POSITION == 4
h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
A = (1./(ALPHA*delta_t) - 1./(delta_x).^2 - 1./(delta_y).^2 -        1./(delta_z).^2 -
h/(delta_z*k))*NODE_TEMPS(T_INDEX);
B =1/(delta_z.^2)*NODE_TEMPS(T1_INDEX) +
        1./(2*delta_y.^2)*NODE_TEMPS(T2_INDEX);
C =  1./(2*delta_x.^2)*NODE_TEMPS(T3_INDEX) + 2*h/(delta_z*k)*Tinf ;
D =  1./(2*delta_y.^2)*NODE_TEMPS(T5_INDEX) +
        1./(2*delta_x.^2)*NODE_TEMPS(T6_INDEX);
TEMP_CNST = A+B+C+D;

%check to see if node is located on side #5.
elseif POSITION == 5
h = Get_h(NODE_TEMPS(T_INDEX),PARAMETERS,POSITION);
A = (1./(ALPHA*delta_t) - 1./(delta_x).^2 - 1./(delta_y).^2 -        1./(delta_z).^2 -
h/(delta_y*k))*NODE_TEMPS(T_INDEX);
B =1/(2*delta_z.^2)*NODE_TEMPS(T1_INDEX) +
        1./(delta_y.^2)*NODE_TEMPS(T2_INDEX);
C =  1./(2*delta_x.^2)*NODE_TEMPS(T3_INDEX) +
        1./(2*delta_z.^2)*NODE_TEMPS(T4_INDEX);
D =  2*h/(delta_y*k)*Tinf + 1./(2*delta_x.^2)*NODE_TEMPS(T6_INDEX);
TEMP_CNST = A+B+C+D;

else
TEMP_CNST = Tinf;

end
```

%%% T 8to5 %%%

```
function Ans = T_8to5(C)

%Following function takes the time temperature history of a node in the form of a vector.
%It removes the heating portion of the history and calculates, with the aid of an
%interpolation function, Δt800-500.

i=4;
delta_t=0.03;
delta_x=0.015;

while (C(i)+1) >= C(i-1)
        i=i+1;
end

if C(i-1)< 1073
        strng = 'This position never reaches 800'
        Ans = -1;
else
        C=C((i-1): size(C,2));
        t_int = (1:size(C,2))*delta_t;

        t8 = interp1(C,t_int,[1073],'spline');
        t5 = interp1(C,t_int,[773],'spline');
        Ans = t5 - t8;
end
```

LIST OF REFERENCES

1. Kou, S., *Welding Metallurgy,* John Wiley and Sons, 1987.

2. Silva, E.A., "Underwater Welding and Cutting," *Metals handbook,* 9th Edition, vol. 6, American Society of Metals, 1983.

3. Johnson, R.J., Master's Thesis, Naval Postgraduate School, Monterey, CA, 1997.

4. Hamann, R. and O. Mahrenholtz, "On the influence of Surface Heat Transfer Coefficient on Wet Underwater Welds," *Proceedings of the Fourth International Offshore and Polar Engineering Conference,* Osaka, Japan, pp. 112-119, April 1994.

5. Tsai, C.L. and K. Masubuchi, "Mechanisms of rapid cooling in underwater welding," *Applied Ocean Research,* Vol. 1, No. 2, 1979.

6. West, T.C., W.E. Mitchell and R.I. Murray, "Effects of Water Temperature on Cracking of Wet-Welded Carbon Steel," *Welding Journal,* Vol. 75, No. 10, pp. 51-55, 1996.

7. Ozisik, M.N., *Finite Difference Methods in Heat Transfer,* CRC Press, Inc., 1994.

8. Callister, W.D., *Materials Science and Engineering,* John Wiley & Sons, Inc., 1994.

9. Lundin, C.D., G. Zhou and K.K. Khan, "Metallurgical Characterization of the HAZ in A516-70 and Evaluation of Fracture Toughness Specimens," *Welding Research Council Bulletin,* No. 403, Report 1, pp. 1-88, July 1995.

10. Brown, R.T. and K. Masubuchi, "Fundamental Research on Underwater Welding," *Welding Journal,* Vol. 54, No. 6, pp. 178s-188s, 1975.

11. Ule, R.L., Y. Joshi and E.B. Sedy, "A New Technique for Three-Dimensional Transient Heat Transfer Computations of Autogenous Arc Welding," *Metallurgical Transactions B,* Vol. 21B, pp. 1033-1047, December 1996.

12. Tsai, C.L. and K. Masubuchi, "Mechanisms of Rapid Cooling and Their Design Considerations in Underwater Welding," *11th Annual Offshore Technology Conference,* OTC 3469, Houston, TX, 1979.

13. Ibarra, S.S. Liu and D.L. Olson, "Underwater Wet Welding of Steels," *Welding Research Council Bulletin,* no. 401, pp. 1-39, May 1995.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Ste 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, California 93943-5101

3. Naval/Mechanical Engineering Curricular Officer, Code 34 1
   Naval Postgraduate School
   Monterey, California 93943-5101

4. Department Chairman, Code ME
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93943-5101 1

5. Dr. Alan G. Fox, Code ME/FX 2
   Mechanical Engineering Department
   Naval Postgraduate School
   Monterey, California 93943-5101

6. Capt. Raymond McCord, SEA OOC
   Supervisor of Salvage and Diving
   Naval Sea Systems Command
   2531 Jefferson Davis Hwy
   Arlington, Virginia 22242-5160 1

7. Mr. Michael Dean, SEA OOC5
   Supervisor of Salvage and Diving
   Naval Sea Systems Command
   2531 Jefferson Davis Hwy
   Arlington, Virginia 22242-5160 1

8. Lt. Jay Dill
   4004 Middleburg Ln.
   Chesapeake, Virginia 23321 1